3.4. Scripts

Definition: Scripts.

- A script is plan of action.
- By a domain script we shall, more specifically, understand:
  - the structured, almost, if not outright,
  - formally expressed, wording of a set of
  - rules and regulations.
- See also:
  - license and
  - contract.

Definitions follow.

3.4.1. Routes as Scripts

3.4.1.1. Paths

92. A path is a triple:

(a) a hub identifier, \( h_i \), a link identifier, \( l_j \), and another hub identifier, \( h_k \), distinct from \( h_i \),

(b) such that there is a link \( \ell \) with identifier \( l_j \) in a net \( n \) such that \( \{h_i, h_k\} \) are the hub identifiers that can be observed from \( \ell \).

Value

93. From a net one can extract all its paths:

(a) if \( l \) is a link of the net,
(b) \( l_j \) its identifier,
(c) \( \{h_i, h_k\} \) the identifiers of its connected hubs,
(d) then \( (h_i, l_j, h_k) \) and \( (h_k, l_j, h_i) \) are paths of the net.

Type

92. \( \text{Pth} = \text{HI} \times \text{LI} \times \text{HI} \)

Axiom

92(a). \( \forall (hi,li,hi) : \text{Pth} \cdot \exists n : \text{N} : l : \text{L} : l_i \in \text{obs}_{\text{Ls}}(n) \Rightarrow \text{92(b). obs}_{\text{LI}}(l) = li \land \text{obs}_{\text{HIs}}(l) = \{hi,hi} \)
94. From a net descriptor one can (likewise) extract all its paths:
   (a) Let \( h_i, h_k \) be any two distinct hub identifiers of the net descriptor (definition set),
   (b) such that they both map into a link identifier \( l_j \),
   (c) then \((h_i, l_j, h_k)\) and \((h_k, l_j, h_i)\) are paths of the net.

value
93. \( \text{paths}: \mathbb{N}D \rightarrow \text{Pth-set} \)
93. \( \text{paths}(\text{nd}) \equiv \)
   94(a). \( \{(h_i, l_j, h_k), (h_k, l_j, h_i)\} \subseteq \text{dom} \text{ nd} \Rightarrow \)
94(b). \( \{l_j \in \text{dom} \text{ nd}(h_i) \cap \text{dom} \text{ nd}(h_k)\} \)

96. From a net, \( n \), we can generate the possibly infinite set of finite and possibly infinite routes:
   (a) \( <> \) is a path (basis clause 1);
   (b) if \( p \) is a path of \( n \) then \( < p > \) is a path of \( n \) (basis clause 2);
   (c) if \( r \) and \( r' \) are non-empty routes of \( n \)
      i. and the last \( h_i \) of \( r \) is the same as the first \( h_j \) of \( r' \)
      ii. then the concatenation of \( r \) and \( r' \) is a route (induction clause).
   (d) Only such routes which can be formed by a (finite, respectively infinite) application of basis clauses Items 96(a) and 96(b) and induction clause Item 96(c) are routes (extremal clause).
3.4.2. Bus Timetables as Scripts

3.4.2.1. Buses

97. Buses are vehicles,
98. with bus identifiers being the same as vehicle identifiers.

\[
\text{type} \quad B \subseteq \text{VI}
\]

3.4.2.2. Bus Stops

99. A link bus stop indicates the link (by its identifier), the from and to hub identifiers, and the fraction “down the link” from the from to the to hub identifiers.

\[
\text{type} \quad \text{BS} = \text{mkL}_{\text{BS}}(\text{sel}_{\text{hi}}:\text{HI},\text{sel}_{\text{li}}:\text{LI},\text{sel}_f:F,\text{sel}_{\text{thi}}:\text{HI})
\]

3.4.2.3. Bus Routes

100. A bus stop list is a sequence of two or more bus stops, \(bsl\).
101. A bus route, \(br\), is a pair of a net route, \(r\), and a bus stop list, \(bsl\), such that route \(r\) is a route of \(n\) and such that \(bsl\) is embedded in \(r\).

(a) there exists an index list, \(il\), of ascending indices of the route \(r\) and of the length of \(bsl\)
(b) such that the \(i\)th path of \(r\)
(c) share from and to hub identifiers and link identifier with the \(il(i)\)th


\[
\text{value} \\
\text{n:N} \\
\text{type} \\
\text{BSL} = \text{BS}^* \\
\text{BR} = \{(r,bsl):(R \times BSL)\text{wf}_{BR}(r,bsl)\}
\]

101(a). \text{is}_\text{embedded}_\text{in}: \text{BR} \rightarrow \text{Bool}
101(a). \text{is}_\text{embedded}_\text{in}(r,bsl) \equiv \exists n:N,r:R \in \text{routes}(n) \land \text{is}_\text{embedded}_\text{in}(r,bsl)

101(b). \exists il:\text{Nat} \cdot \text{len (il)} = \text{len} \text{bsl} \land \text{il} \subseteq \text{inds} r \land \text{ascending}(il) \Rightarrow
101(c). \forall i:\text{Nat} \cdot i \in \text{inds} il \Rightarrow
101(c). \begin{cases}
\text{let (hi, lj, hk)} = r(il(i)), (hi', lj', fhk) = bsl(i) \text{ in} \\
hi = hi' \land lj = lj' \land hk = hk' \text{ end} \land \\
\forall i:\text{Nat} \cdot \{i,i+1\} \subseteq \text{inds} il \Rightarrow
102. \text{let (hi, lj, fhk)} = bsl(i), (hi', lj', fhk') = bsl(i+1) \text{ in} \\
hi = hi' \land lj = lj' \land fhk = fhk' \Rightarrow f < f'
\end{cases}
\]

ascending: \text{Nat}^* \rightarrow \text{Bool}, \text{ascending}(il) \equiv \forall i:\text{Nat} \cdot \{i,i+1\} \subseteq \text{inds} il \Rightarrow il(i) < il(i+1)

3.4.2.4. Bus Schedule

103. A timed bus stop is a pair of a time and a bus stop.
104. A timed bus stop list is a sequence of timed bus stops.
105. A bus schedule is a pair of a route and a timed bus stop list such that

- there is a net of which the routes is indeed a route,
- the bus stop list of the timed bus stop list is embedded in the route, and
- ‘later’ listed bus stops register later times.

106. SimpleBusSchedules remove routes from BusRoutes.
The concept of a bus line captures all those bus schedules which ply the same bus route but at different times. A timetable is made up from distinctly named bus lines.

107. A bus line has a unique bus line name.
108. We say that two bus schedules are the same if they are based on the same route and if they differ only in their times.
109. Each of the different bus routes of a bus line has a unique bus number.
110. A route bus schedule pairs a route with simple bus schedules for each of a number of busses (identified by their bus number).
111. A bus timetable (listing, map) maps bus line names to route bus schedules.
112. A timetable is a pair, a net and a table.
113. A well-formed timetable must satisfy same bus schedules within each bus line.
114. All bus numbers are distinct across bus lines.
3.4.3. Route and Bus Timetable Denotations

- What are routes and bus timetables scripting?

- Routes (list of connected link traversal designations) script that one may transport people or freight along the sequence of designated links.

- Bus timetables script (at least) two things:
  - the set of bus traffics on the net which satisfy the bus timetable, and
  - information that potential and actual bus passengers may, within some measure of statistics (and probability), rely upon for their bus transport.

3.4.4. Licenses and Contracts

**Definition: License.**

- A license is
  - a *script*
  - specifically expressing a permission to act;
  - is freedom of action;
  - is a permission granted by competent authority to engage in a business or occupation or in an activity otherwise unlawful;
  - a document, plate, or tag evidencing a license granted;
  - a grant by the holder of a copyright or patent to another of any of the rights embodied in the copyright or patent short of an assignment of all rights.

Licenses appear more to have morally than legally binding poser.

- Here, we shall not develop the idea of bus timetables denoting certain traffics.
  - Instead we refer to our previously sketched model of traffics (Sect., Pages 89-97).

- Route (designations) and bus timetables
  - script potential and actual route travels, respectively
  - script the dispatch of buses and their travelling.

- Bus timetables can also be seen as a form of contracts
  - between the bus operators offering the bus services
  - and potential and actual passengers,
  - with the contract promising timely transport.

- In the next section, Sect., we shall sketch a language of bus service contracts and bus service actions implied by such contracts.

**Definition: Contract.**

- A contract
  - is a special kind of license
  - specifically expressing a legally binding agreement between two or more parties —
  - hence a document describing the conditions of the contract;
  - a contract is business arrangement for the supply of goods or services at fixed prices, times and locations.
• In software development a contract specifies what is to be developed:
  – (1) a domain description,
  – (2) a requirements prescription, or
  – (3) a software design;

or a combination of these (1–2, 2–3, 1–3).

• A contract further specifies
  – how it might, or must be developed;
  – criteria for acceptance of what has been developed;
  – delivery dates for the developed items;
  – who the “parties” to the contract are:
    * the client and
    * the developer, etc.

• For a comprehensive treatment of licenses and contracts we refer to [Chapter 10, Sect. 10.6 (Pages 309–326)](jaist-db10)[jaist-mono].

• We shall illustrate fragments of a language for bus service contracts.

• The background for the bus contract language is the following.
  – In many large cities around Europe the city or provincial government secures public transport in the form of bus services operated by many different private companies.
  – Earlier lectures illustrated the concept of bus (service) timetables.
  – The bus services implied by such a timetable, for a city area — with surrounding suburbs etc. — need not be implemented by just one company, but can be contracted, by the city government public transport office, to several companies, each taking care of a subset of the timetable.

  – Different bus operators then take care of non-overlapping parts and all take care of the full timetable.
  – It may even be that extra buses need be scheduled, on the fly, in connection with major sports or concert or other events.
  – Bus operators may experience vehicle breakdowns or bus driver shortages and may be forced to subcontract other, even otherwise competing bus operators to “step in” and alleviate the problem.
3.4.4.1. Contracts

Schematically we may represent a bus contract as follows:

Contract \( cn \) between contractee \( ci \) and contractor \( cj \):

This contract contracts \( cj \) in the period \( [t,t'] \) to perform the following services with respect to timetable \( tt \):

- operate bus lines \( \{blj_1, blj_2, ..., blj_n\} \)
- subject to the following occasional exceptions:
  - cancellation of bus tours:
    \( \{\{blj_{no_1}, blj_{no_2}, ..., blj_{no_m}\}\} \) subject to conditions \( cbt \)
  - insertion of bus tours on lines
    \( \{blj_1, blj_2, ..., blj_j\} \) subject to conditions \( ibt \)
  - subcontracting bus tours on lines
    \( \{blj_1, blj_2, ..., blj_m\} \) subject to conditions \( scbt \).

We abstract the above quoted “one or more of three kinds of exceptions” as one possibly empty clause for each of these alternatives.

115. A bus contract has a header with the distinct names of a contractee and a contractor and a time interval.

116. A bus contract presents a timetable.

117. A bus contract presents a set of bus lines (by their identifiers) such that these are in the timetable.

118. And a bus contract may list one or more of three kinds of “exceptions”:

(a) cancellation of one or more named bus tours on one or more bus lines subject to certain (specified) conditions;
(b) insertion of one or more extra bus tours on one or more bus lines subject to certain (specified) conditions;
(c) subcontracting one or more unspecified bus tours on one or more bus lines subject to certain (specified) conditions — to further unspecified contractors.

119. A bus contract now contains a header, a timetable, the subject bus lines and the exceptions,

120. such that

(a) line names mentioned in the contract are those of the bus lines of the timetable, and

(b) bus (tour) numbers are those of the appropriate bus lines in the timetable.

121. The calendar period is for at least one full day, midnight to midnight.

122. A named contract is a pair of a contract name and a contract.

\[ \text{type} \]

115. \( CNm, Clid, D, T, CON \)
115. \( CH = Clid \times Clid \times (D \times D) \)
116. \( CT = TT \)
117. \( CLs = BLNm-set \)
118. \( CE = (CA \times IN \times SC) \times CON \)
118(a). \( CA = BLNm \rightarrow \text{BNo-set} \)
118(b). \( IN = BLNm \rightarrow \text{BNo-set} \)
118(c). \( SC = BLNu-set \)
119. \( CO = CH \times CT \times CLs \times CE \)
120. \( CO = \{\{co: CO\text{-}wf, CO(co)\}\} \)
122. \( NCO = CNm \times CO \)
value
120. $\text{wf}_C: C \rightarrow \text{Bool}$
120. $\text{wf}_C((\text{ce},\text{cr},(d,d)),(\text{nd},\text{tbl}),\text{cls},((\text{blns},\text{blns},\text{bli}),\text{con})) \equiv$
117. $\text{ce} \neq \text{cr} \land$
120(a). $\text{cls} \subseteq \text{dom} \text{ tbl} \land$
120(b). $\forall \text{bli},\text{bli}':\text{BLNm} \cdot \text{bli} \in \text{dom} \text{ blns} \land \text{bli}' \in \text{dom} \text{ blns'} \Rightarrow$
120(a). $\{\text{bli},\text{bli}'\} \subseteq \text{dom} \text{ tbl} \land$
120(b). $\text{blns}(\text{bli}) \cup \text{blns}(\text{bli}') \subseteq \text{dom} \text{ sel}_b \text{ tbl}(\text{tbl}(\text{bli})) \land$
120(a). $\text{bli} \subseteq \text{dom} \text{ tbl} \land$
121. $d < d'$

123. A bus operator action is either a commence, a cancellation, an insertion or a subcontracting action. All actions refer to the (name of) the contract with respect to which the action is contracted.

(a) A commence action designator states the bus line concerned and the bus number of that line.

(b) A cancellation action designator states the bus line concerned and the bus number of that line.

(c) An insertion action designator states the bus line concerned and the bus number of that line — for which an extra bus is to be inserted.\(^4\)

(d) A subcontracting action designator, besides the name of the contract with respect to which the subcontract is a sub caveat (whose contract name is unique).

\(^4\) The insertion of buses in connection with either unscheduled or extraordinary (sports, concerts, etc.) events can be handled by special, initial contracts.
3.4.4.3. Wellformedness of Contractual Actions

124. In order to express wellformedness conditions, that is, pre-conditions, for the action designators we introduce a context which map contract names to contracts.

125. Wellformedness of a contract is now expressed with respect to a context.

\[ \text{type} \]
\[ \text{CTX} = \text{CNm} \rightarrow \text{CO} \]

\[ \text{value} \]
\[ \text{wf}_\text{Act}: \text{Act} \rightarrow \text{CTX} \rightarrow \text{Bool} \]

127. cancellation and insertion commands have the same static wellformedness conditions as have commence command.

127. \[ \text{wf}_\text{Act}(\text{mkCan}(\text{cnm}, \text{bln}, \text{bno}))(\text{ctx}) \equiv \text{wf}_\text{Act}(\text{mkCom}(\text{cnm}, \text{bln}, \text{bno}))(\text{ctx}) \]

127. \[ \text{wf}_\text{Act}(\text{mkIns}(\text{cnm}, \text{bln}, \text{bno}))(\text{ctx}) \equiv \text{wf}_\text{Act}(\text{mkCom}(\text{cnm}, \text{bln}, \text{bno}))(\text{ctx}) \]

- Let a defined \( \text{cnm} \) entry in \( \text{ctx} \) be a contract: \((\text{ce}, \text{cr}, (\text{nd}, \text{tbl}), \text{cls}, (\text{blns}, \text{bls}, \text{bls}''), (\text{d}, \text{d}'))\).

126. If \( \text{cmd} \) is a commence command \( \text{mkCom}(\text{cnm}, \text{bln}, \text{bno}) \), then

(a) contract name \( \text{cnm} \) must be defined in context \( \text{ctx} \);

(b) bus line name \( \text{bln} \) must be defined in the contract, that is, in \( \text{cls} \), and

(c) bus number \( \text{bno} \) must be defined in the bus table part of table \( \text{tbl} \).

126. \[ \text{wf}_\text{Act}(\text{mkCom}(\text{cnm}, \text{bln}, \text{bno}))(\text{ctx}) \equiv \]

126(a). \( \text{cnm} \in \text{dom} \text{ctx} \land \]

126(b). \( \text{let } ((\text{ce}, \text{cr}, (\text{nd}, \text{tbl}), \text{cls}, (\text{blns}, \text{bls}, \text{bls}')), (\text{d}, \text{d}')) = \text{ctx}(\text{cnm}) \text{ in} \]

126(c). \( \text{bno} \in \text{dom} \text{sel}_\text{btbl}(\text{tbl}(\text{bln})) \text{ end} \]

128. If \( \text{cmd} \) is a subcontract command then

Let the subcontract command and the \( \text{cnm} \) named contract in \( \text{ctx} \) be

\[ \text{mkSub}(\text{cnm}, \text{nco}:(\text{cnm}', (\text{ce}', \text{cr}', (\text{d}', \text{d}''))), (\text{nd}', \text{tbl}'), \text{cls}', (\text{blns}', \text{bls}', \text{bls}''))\)

respectively \((\text{ce}, \text{cr}, (\text{d}, \text{d}'))\), \((\text{nd}, \text{tbl})\), \text{cls}, (\text{blns}, \text{bls}, \text{bls}')\).

(a) contract name \( \text{cnm} \) must be defined in context \( \text{ctx} \);

(b) contract name \( \text{cnm}' \) must not be defined in context \( \text{ctx} \);

(c) the calendar period of the subcontract must be within that of the contract from which it derives;

(d) the net descriptors \( \text{nd} \) and \( \text{nd}' \) must be identical;

(e) the tables \( \text{tbl} \) and \( \text{tbl}' \) must be identical and

(f) the set, \( \text{cls}' \), of bus line names that are the scope of the subcontracting must be a subset of \( \text{bls}' \).
3.4.4.3. Wellformedness of Contractual Actions

- Wellformedness of contracts, $\text{wf}_{\text{CO}}(co)$ and $\text{wf}_{\text{CO}}(co')$, secures other constraints.

\begin{verbatim}
128. $\text{wf}\_\text{Act}(\text{mkSub}(\text{cmn},\text{ncm}\cdot\text{co}::((\text{ce}',\text{cr}',(\text{d}',\text{d}'')),(\text{nd}',\text{tbl}'),\text{cls}'\cdot(\text{blns}',\text{blns}',\text{bls}')))))(\text{ctx})$
128(a). $\text{cmn} \in\text{dom} \text{ ctx} \land$
128(b). $\text{let} \text{ co}' = ((\text{ce},\text{cr},(\text{d},\text{d}')),(\text{nd},\text{tbl}),\text{cls},(\text{blns},\text{blns},\text{bls})) = \text{ctx}(\text{cmn})$ in
128(c). $\text{cmn}' \notin\text{dom} \text{ tbl} \land$
128(d). $\text{d} \leq \text{d}'' \leq \text{d}''' \leq \text{d}' \land$
128(e). $\text{nd}' = \text{nd} \land$
128(f). $\text{tbl}' = \text{tbl} \land$
128(g). $\text{cls}' \subseteq \text{bls}'$ end
\end{verbatim}

- We do not here bring any narrated or formalised description of the semantics of contracts and actions.
- First such a description would be rather lengthy.
- Secondly a specification would be more of a requirements prescription.

3.5. Management and Organisation

**Definition: Management.**

- **Management is about resources:**
  - their **acquisition**,
  - **scheduling** (over time),
  - **allocation** (over locations),
  - deployment (in performing actions) and
  - disposal (“retirement”).

- **We distinguish between**
  - **board-directed**,  
  - **tactical and**
  - **strategic**,  
  - **operational**
  
  **actions.**

- **Board-directed** actions target mainly financial resources: obtaining new funds through conversion of goodwill into financial resources, acquiring and selling “competing” or “supplementary” business units.
- **Strategic** actions convert financial resources into production, service supplies and resources and vice-versa — and in this these actions schedule availability of such resources.
- **Tactical** actions mainly allocate resources.
- **Operational** actions order, monitor and control the deployment of resources in the performance of actions.
Definition: Organisation.

- **Organisation is about**
  - the “grand scale”,
    - executive and strategic
    - national, continental or global (world wide)
  - (i) *allocation* of major resource (e.g., business) units, whether in a hierarchical, in a matrix, or in some other organigram-specified structure,
  - (ii) as well as the clearly defined *relations* (which information, decisions and actions are transferred) between these units, and
  - (iii) organisational *dynamics*.

**Definition: Management & Organisation.**

- The composite term *management* and *organisation* — applies in connection with *management* as outlined just above and with *organisation* also outlined above.
- The term then emphasises the relations between the organisation and management of an enterprise.

The borderlines within management actions and across organisation “layouts” are fuzzy.

### 3.5.1. Transport System Examples

We shall only present sketchy examples of *management* and *organisation*.

- **Executive actions:**
  - Deciding on major re-organisation of a transport net
    - (for example introduction of toll roads or freeways,
      - road pricing,
      - major bridges across wide waters [potentially connecting two hitherto unconnected nets],
      - and their management)
    - are executive actions.
  - So are decisions on merging or splitting transport from or into several transport services.

- Reorganising an enterprise
  - from one characterised by a “deep” hierarchy of management layers (a hierarchy which may very well exemplify highly centralised both administrative and functional monitoring and control)
    - into a matrix of two “shallow” hierarchies, one which addresses tactical and operational management and one which addresses executive and strategic management — with the former (the operations) being replicated across geographical areas while the latter applies “globally” —
  - such reorganisations reflect executive actions (but are carried out by strategic and tactical management).
3. Management and Organisation

3.5. Management and Organisation

3.5.1. Transport System Examples

• Strategic actions: Adding or removing transport links, or major reorganisation of bus timetables are strategic actions. Splitting a(n own) contract into what is still to be operated and subcontracting other parts, for definite, to other bus operators are also strategic actions.

• Tactical actions: Insertion and cancellation of bus services are tactical actions. Subcontracting some parts of a timetable demanded service, for a short while, to other bus operators could be considered tactical actions.

• Operational actions: Commencing and thus, in general, allocating drivers to and sending these off on bus services are operational actions. So are announcing insertion of new (unscheduled) and cancellation of scheduled routes.

3.6. Human Behaviour

Definition: Human Behaviour.

• By human behaviour we shall here understand
  - the way a human follows the enterprise rules and regulations
  - as well as interacts with a machine:
    * dutifully honouring specified (machine dialogue or) protocols,
    * or negligently so,
    * or sloppily not quite so,
    * or even criminally not so!

• Human behaviour is a facet of the domain.
  - We shall thus model human behaviour also in terms of it failing to react properly,
  - i.e., humans as non-deterministic agents!

3.7. Towards Theories of Domain Facets
3.7.1. A Theory of Intrinsics

- For all intrinsic traffics, \( \text{itf} \), and for all optical sensor technologies, \( \text{og} \), the following must hold:
  - Let \( \text{stf} \) be the traffic sampled by the optical gates.
  - For all time points, \( t \), in the sampled traffic,
  - those time points must also be in the intrinsic traffic,
  - and, for all trains, \( \text{tn} \), in the intrinsic traffic at that time,
  - the train must be observed by the optical gates, and
  - the actual position of the train and the sampled position must somehow be checkable to be close, or identical to one another.

Since hubs change state with time, \( n: \mathbb{N} \), the net needs to be part of any model of traffic.

3.7.2. Theories of Support Technologies

3.7.2.1. An Example

- Traffic (\( \text{tf}: \text{TF} \)), intrinsically, is a total function over some time interval, from time (\( t:T \)) to continuously positioned (\( p:P \)) vehicles (\( \text{tn}: \text{TN} \)).
- Conventional optical sensors sample, at regular intervals, the intrinsic train traffic.
- The result is a sampled traffic (\( \text{stf}: \text{sTF} \)).
- Hence the collection of all optical sensors, for any given net, is a partial function from intrinsic (\( \text{itf} \)) to sampled train traffics (\( \text{stf} \)).
- We need to express quality criteria that any optical sensor technology should satisfy — relative to a necessary and sufficient description of a closeness predicate.

\[
\begin{align*}
\text{type} & \quad T, \text{TN} \\
& \quad P = \text{HP} \ | \ \text{LP} \\
\text{NetTraffic} & :: \text{net}: \mathbb{N} \times \text{trf}:(V \rightarrow m \text{P}) \\
\text{iTF} & = T \rightarrow \text{NetTraffic} \\
\text{sTF} & = T \rightarrow m \text{NetTraffic} \\
\text{oG} & = \text{iTF} \sim \text{sTF} \\
\text{value} & \quad \star \\n\text{axiom} & \quad \forall \text{itt}: \text{iTF}, \text{og}: \text{OG}. \quad \text{let} \quad \text{stt} = \text{og}(\text{itt}) \quad \text{in} \\
& \quad \forall t:T. \quad t \in \text{DOM} \text{stt} \cdot \\
& \quad \text{t} \in \text{DOM} \quad \text{itt} \quad \forall \text{Tn}: \text{TN}. \quad \text{tn} \in \text{DOM} \quad \text{trf}(\text{itt}(t)) \\
& \quad \Rightarrow \quad \text{tn} \in \text{DOM} \quad \text{trf}(\text{stt}(t)) \quad \land \quad c(\text{itt}(t), \text{tn}, \text{stt}(t)) \quad \text{end}
\end{align*}
\]
3.7.2.2. General

- The formal requirements can be narrated:
  - Let $\Theta_i$ and $\Theta_a$ designate the spaces of intrinsic and actual-world configurations (contexts and states).
  - For each intrinsic configuration model — that we know is support technology assisted —
  - there exists a support technology solution,
  - that is, a total function from all intrinsic configurations to corresponding actual configurations.
- If we are not convinced that there is such a function then there is little hope that we can trust this technology.

**type**

$\Theta_i, \Theta_a$

$ST = \Theta_i \rightarrow \Theta_a$

**axiom**

$\forall \text{sts}:\text{ST-set}, \text{st}:\text{ST} \cdot \text{st} \in \text{sts} \Rightarrow \forall \theta_i:\Theta_i, \exists \theta_a:\Theta_a \cdot \text{st}(\theta_i) = \theta_a$

3.7.3. A Theory of Rules & Regulations

- There are, abstractly speaking, usually three kinds of languages involved wrt. (i.e., when expressing) rules and regulations (respectively when invoking actions that are subject to rules and regulations).
  - Two languages, Rules and Reg, exist for describing rules, respectively regulations; and
  - one, Stimulus, exists for describing the form of the [always current] domain action stimuli.

**type**

Stimulus, Rule, $\Theta$

$\text{STI} = \Theta \rightarrow \Theta$

$\text{RUL} = (\Theta \times \Theta) \rightarrow \text{Bool}$

**value**

meaning: Stimulus $\rightarrow$ STI

meaning: Rule $\rightarrow$ RUL

valid: Stimulus $\times$ Rule $\rightarrow$ $\Theta \rightarrow$ Bool

valid$(\text{sy}_{\text{sti}}, \text{sy}_{\text{rul}})(\theta) \equiv \text{meaning}(\text{sy}_{\text{rul}})(\theta, (\text{meaning}(\text{sy}_{\text{sti}})))(\theta)$

valid: Stimulus $\times$ RUL $\rightarrow$ $\Theta \rightarrow$ Bool

valid$(\text{sy}_{\text{sti}}, \text{se}_{\text{rul}})(\theta) \equiv \text{se}_{\text{rul}}(\theta, (\text{meaning}(\text{sy}_{\text{sti}})))(\theta)$
3. An Ontology of Domain Facets

3.7. Towards Theories of Domain Facets

3.7.3. A Theory of Rules & Regulations

- A syntactic regulation, $\text{sy}_\text{reg}:\text{Reg}$ (related to a specific rule), stands for, i.e., has as its semantics, its meaning,
  - a semantic regulation, $\text{se}_\text{reg}:\text{REG}$,
  - which is a pair.
  - This pair consists of
    * a predicate, $\text{pre}_\text{reg}:\text{Pre}_\text{REG}$, where $\text{Pre}_\text{REG} = (\Theta \times \Theta) \rightarrow \text{Bool}$,
    * and a domain configuration-changing function, $\text{act}_\text{reg}:\text{Act}_\text{REG}$, where $\text{Act}_\text{REG} = \Theta \rightarrow \Theta$,
    * that is, both involving current and next domain configurations.

- The two kinds of functions express:
  * If the predicate holds,
  * then the action can be applied.

- The predicate is almost the inverse of the rules functions.
- The action function serves to undo the stimulus function.

- The idea is now the following:
  - Any action of the system, i.e., the application of any stimulus,
    * may be an action in accordance with the rules,
    * or it may not.
  - Rules therefore express whether stimuli are valid or not in the current configuration.
  - And regulations therefore express whether they should be applied, and, if so, with what effort.
• More specifically,
  – there is usually, in any current system configuration, given a set of pairs of rules and regulations.
  – Let \( (\text{sy}_\text{rul}, \text{sy}_\text{reg}) \) be any such pair.
  – Let \( \text{sy}_\text{sti} \) be any possible stimulus.
  – And let \( \theta \) be the current configuration.
  – Let the stimulus, \( \text{sy}_\text{sti} \), applied in that configuration result in a next configuration, \( \theta' \), where \( \theta' = (\text{meaning}(\text{sy}_\text{sti}))(\theta) \).
  – Let \( \theta' = (\text{meaning}(\text{sy}_\text{sti}))(\theta) \) violate the rule, i.e., \( \neg \text{valid}(\text{sy}_\text{sti}, \text{sy}_\text{rul})(\theta) \)
  – then if predicate part, \( \text{pre}_\text{reg} \), of the meaning of the regulation, \( \text{sy}_\text{reg} \), holds in that violating next configuration, \( \text{pre}_\text{reg}(\theta, \theta') \)
  – then the action part, \( \text{act}_\text{reg} \), of the meaning of the regulation, \( \text{sy}_\text{reg} \), must be applied, \( \text{act}_\text{reg}(\theta') \), to remedy the situation.

axiom
\[
\forall (\text{sy}_\text{rul}, \text{sy}_\text{reg}) : \text{Rul}_\text{and}_\text{Regs} . \\
\text{let } \text{se}_\text{rul} = \text{meaning}(\text{sy}_\text{rul}), \\
(\text{pre}_\text{reg}, \text{act}_\text{reg}) = \text{meaning}(\text{sy}_\text{reg}) \text{ in} \\
\forall \text{sy}_\text{sti} : \text{Stimulus}, \theta : \Theta . \\
\neg \text{valid}(\text{sy}_\text{sti}, \text{se}_\text{rul})(\theta) \\
\Rightarrow \text{let } \theta' = (\text{meaning}(\text{sy}_\text{sti}))(\theta) \text{ in} \\
\text{pre}_\text{reg}(\theta, \theta') \\
\Rightarrow \exists n \theta : \Theta . \text{act}_\text{reg}(\theta') = n \theta \land \text{se}_\text{rul}(\theta, n \theta)
\]

• It may be that the regulation predicate fails to detect applicability of regulations actions.
• That is, the interpretation of a rule differs, in that respect, from the interpretation of a regulation.
• Such is life in the domain, i.e., in actual reality
3.7.5. A Theory of Human Behaviour

- Commensurate with the above, humans interpret rules and regulations differently,
- and not always “consistently” — in the sense of repeatedly applying the same interpretations.
- Our final specification pattern is therefore:

\[ \text{Action} = \Theta \xrightarrow{\sim} \Theta-\text{infset} \]

\[ \text{hum}_i: \text{Rule} \xrightarrow{\sim} \Theta \xrightarrow{\sim} \text{RUL-\text{infset}} \]

\[ \text{action}: \text{Stimulus} \xrightarrow{\sim} \Theta \xrightarrow{\sim} \Theta \]

\[ \text{hum}_b: \text{Stimulus} \times \text{Rules} \xrightarrow{\sim} \Theta \xrightarrow{\sim} \Theta-\text{infset} \]

\[ \text{hum}_b(sy_{sti},sy_{rul})(\alpha)() \xrightarrow{\sim} \Theta-\text{infset} \]

\[ \text{post} \]

\[ \theta \text{set} = \alpha(\theta) \land \text{action}(sy_{sti})(\theta) \in \theta \text{set} \]

\[ \land \forall \theta' : \theta \theta' \in \theta \text{set} \Rightarrow \exists \text{se}_\text{rul}: \text{RUL-\text{infset}} \in \text{hum}_i(sy_{rul})(\theta) \Rightarrow \text{se}_\text{rul}(\theta,\theta') \]

- The above is, necessarily, sketchy:
  - There is a possibly infinite variety of ways of interpreting some rules.
  - A human, in carrying out an action, interprets applicable rules and chooses one which that person believes suits some (professional, sloppy, delinquent or criminal) intent.
  - “Suits” means that it satisfies the intent,
    * i.e., yields true on the pre/post-configuration pair,
    * whether as intended by the ones who issued the rules and regulations or not.
  - We do not cover the case of whether an appropriate regulation is applied or not

End of Lecture 4: DOMAINS: Scripts – Human Behaviour
B  Slide Table-of-Contents

Contents

<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>1 Introduction</td>
<td>5</td>
</tr>
<tr>
<td>1.1 The Problem</td>
<td>5</td>
</tr>
<tr>
<td>1.2 The Triptych Approach</td>
<td>6</td>
</tr>
<tr>
<td>2 An Ontology of Specification Entities</td>
<td>9</td>
</tr>
<tr>
<td>2.1 Simple Entities</td>
<td>11</td>
</tr>
<tr>
<td>2.1.1 Net, Hubs and Links</td>
<td>13</td>
</tr>
<tr>
<td>2.1.2 Unique Hub and Link Identifiers</td>
<td>14</td>
</tr>
<tr>
<td>2.1.3 Observability of Hub and Link Identifiers</td>
<td>15</td>
</tr>
<tr>
<td>2.1.4 A Theorem</td>
<td>17</td>
</tr>
<tr>
<td>2.1.5 Link implies Hub</td>
<td>18</td>
</tr>
<tr>
<td>2.1.6 Hub and Link Attributes</td>
<td>19</td>
</tr>
<tr>
<td>2.2 States</td>
<td>22</td>
</tr>
<tr>
<td>2.3 Actions</td>
<td>23</td>
</tr>
<tr>
<td>2.3.1 Insert Hubs</td>
<td>23</td>
</tr>
<tr>
<td>2.3.2 Remove Hubs</td>
<td>25</td>
</tr>
<tr>
<td>2.3.3 Insert Links</td>
<td>27</td>
</tr>
<tr>
<td>2.3.4 Remove Links</td>
<td>31</td>
</tr>
<tr>
<td>2.3.5 Two Theorems</td>
<td>34</td>
</tr>
<tr>
<td>2.3.6 Idempotency</td>
<td>34</td>
</tr>
<tr>
<td>2.3.7.1 Reachability</td>
<td>35</td>
</tr>
<tr>
<td>2.4 Events</td>
<td>37</td>
</tr>
<tr>
<td>2.5 Behaviours</td>
<td>41</td>
</tr>
<tr>
<td>2.5.1 Behaviour Prescriptions</td>
<td>42</td>
</tr>
<tr>
<td>2.5.1.1 Construction Plans</td>
<td>42</td>
</tr>
<tr>
<td>2.3.3.4 Another Traffic Regulator</td>
<td>39</td>
</tr>
<tr>
<td>2.3.4 Remove Links</td>
<td>107</td>
</tr>
<tr>
<td>2.3.4.1 Buses</td>
<td>107</td>
</tr>
<tr>
<td>2.3.4.1 Path</td>
<td>107</td>
</tr>
<tr>
<td>2.3.4.1.1 Paths</td>
<td>107</td>
</tr>
<tr>
<td>2.3.4.2 Bus Timetables as Scripts</td>
<td>107</td>
</tr>
<tr>
<td>2.3.4.2 Buses</td>
<td>107</td>
</tr>
<tr>
<td>2.3.4.2.1 Buses</td>
<td>107</td>
</tr>
<tr>
<td>2.3.4.2.2 Bus Stops</td>
<td>107</td>
</tr>
<tr>
<td>2.3.4.2.3 Bus Routes</td>
<td>108</td>
</tr>
<tr>
<td>2.3.4.2.4 Bus Schedule</td>
<td>110</td>
</tr>
<tr>
<td>2.3.4.2.5 Timetable</td>
<td>112</td>
</tr>
<tr>
<td>2.3.4.3 Route and Bus Timetable Denotations</td>
<td>115</td>
</tr>
<tr>
<td>2.3.4.4 Licenses and Contracts</td>
<td>117</td>
</tr>
<tr>
<td>2.3.4.4.1 Contracts</td>
<td>117</td>
</tr>
<tr>
<td>2.3.4.4.2 Contractual Actions</td>
<td>123</td>
</tr>
<tr>
<td>2.3.4.4.2.1 Contractual Actions</td>
<td>128</td>
</tr>
<tr>
<td>2.3.4.4.3 Wellformedness of Contractual Actions</td>
<td>131</td>
</tr>
<tr>
<td>2.3.4.5 Management and Organisation</td>
<td>137</td>
</tr>
<tr>
<td>2.3.4.5.1 Transport System Examples</td>
<td>141</td>
</tr>
<tr>
<td>2.3.4.5.2 General</td>
<td>146</td>
</tr>
<tr>
<td>2.3.4.5.2.1 General</td>
<td>146</td>
</tr>
<tr>
<td>2.3.4.5.2.2 General</td>
<td>146</td>
</tr>
<tr>
<td>2.3.4.5.3 A Theory of Intrinsics</td>
<td>147</td>
</tr>
<tr>
<td>2.3.4.5.3.1 A Theory of Intrinsics</td>
<td>147</td>
</tr>
<tr>
<td>2.3.4.5.3.2 A Theory of Support Technologies</td>
<td>148</td>
</tr>
<tr>
<td>2.3.4.5.3.2.1 An Example</td>
<td>148</td>
</tr>
<tr>
<td>2.3.4.5.3.2.2 General</td>
<td>151</td>
</tr>
<tr>
<td>2.3.4.5.3.3 A Theory of Rules &amp; Regulations</td>
<td>152</td>
</tr>
<tr>
<td>2.3.4.5.3.4 A Theory of Management &amp; Organisation</td>
<td>162</td>
</tr>
<tr>
<td>2.3.4.5.3.5 A Theory of Human Behaviour</td>
<td>163</td>
</tr>
</tbody>
</table>

4 An Ontology of Requirements Constructions

4.1 Business Process Re-engineering | 166 |
| 4.1.1 The Kinds of Requirements | 169 |
| 4.1.2 Goals Versus Requirements | 172 |
| 4.1.2.1 Goals of a Toll Road System | 175 |

4.2 Domain Requirements

4.2.1 Example | 193 |
| 4.2.1.1 Example | 193 |
| 4.2.2 Abstraction | 196 |
| 4.2.2.1 Example | 197 |
| 4.2.2.2 Abstraction: From Concrete Toll Road Nets to Abstract Nets | 202 |
| 4.2.2.3 Theorems | 203 |
| 4.2.3 Determination | 204 |
| 4.2.3.1 Example | 205 |

4.3 Interface Requirements

4.3.1 But First: On Shared Phenomena and Concepts | 235 |
| 4.3.1.1 Example | 239 |
| 4.3.1.2 Example | 240 |
| 4.3.3 Shared Actions | 240 |
| 4.3.3.1 Example | 241 |
| 4.3.3.2 Example | 242 |

4.4 Requirements Constructions

4.4.1 Examples | 263 |
| 4.4.1.1 Examples | 263 |
| 4.4.1.2 Examples | 263 |

4.5.1.2 Wellformedness of Construction Plans | 43 |
| 4.5.2 Augmented Construction Plans | 47 |
| 4.5.3 Sequential Construction Behaviours | 50 |

Lect. # 3: DOMAINS: Intrinsics – Rules & Regulations

3 An Ontology of Domain Facets

3.0 Definitions | 53 |
| 3.0.1 Definitions | 53 |
| 3.0.2 Business Processes | 58 |
| 3.0.2.1 A Characterisation | 58 |
| 3.0.2.2 An Example | 58 |
| 3.1 Intrinsics | 59 |
| 3.1.1 Net Topology Descriptors | 60 |
| 3.1.2 Link States and Link State Spaces | 63 |
| 3.1.3 Hub States and Hub State Spaces | 67 |
| 3.1.4 State and State Space Wellformedness | 69 |
| 3.1.5 Concrete Types for Simple Entities | 70 |
| 3.1.6 Example Hub Crossings | 73 |
| 3.1.7 Actions Continued | 75 |
| 3.2 Support Technologies | 77 |
| 3.2.1 Traffic Signals | 78 |
| 3.2.2 Traffic “Control” | 82 |
| 3.3 Rules and Regulations | 84 |
| 3.3.1 Vehicles | 87 |
| 3.3.2 Traffic | 89 |
| 3.3.2.1 Wellformedness of Traffic | 89 |
| 3.3.2.1.1 Static Wellformedness | 90 |
| 3.3.2.1.2 Dynamic Wellformedness | 92 |
| 3.3.3 Traffic Rules (II of II) | 97 |
| 3.3.4 Another Traffic Regulator | 98 |
| 3.3.5 Traffic Rules (II of II) | 99 |